Today we are going to discuss implementation of the simplest possible partial differential equations:

\[
\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0
\]

Discretized with the finite volume scheme:

\[
\bar{\rho}_i^{n+1} = (1 - \lambda) \bar{\rho}_i^n + (\lambda) \bar{\rho}_{i-1}^n
\]

Where:

\[
\bar{\rho}_i^n \approx \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} \rho(x, t = n dt) dx
\]

and:

\[
\lambda = \bar{u} \frac{dt}{dx}
\]
Serial Implementation

• The recurrence relationship: $\bar{\rho}_i^{n+1} = (1 - \lambda) \bar{\rho}_i^n + (\lambda) \bar{\rho}_{i-1}^n$

• Holds for $i=1,2,\ldots,N-1$ and $n=0,1,\ldots,M$

• Where $N = \# \text{ of cells discretizing the pipe}$

• and $M = \# \text{ of time steps to be taken}$.

• We will need some storage..
Storage

• Given the scheme:  \[ \overline{\rho}_{i}^{n+1} = (1 - \lambda) \overline{\rho}_{i}^{n} + (\lambda) \overline{\rho}_{i-1}^{n} \]

• We note that to compute the (n+1)’th time level we only need data from the n’th time level.

• For ease of coding let us create two vectors of doubles:  
  \[ \overline{\sigma} = \left( \overline{\rho}_{0}^{n+1}, \overline{\rho}_{1}^{n+1}, \overline{\rho}_{2}^{n+1}, \overline{\rho}_{3}^{n+1}, \ldots, \overline{\rho}_{N-1}^{n+1} \right) \]
  \[ \overline{\rho} = \left( \overline{\rho}_{0}^{n}, \overline{\rho}_{1}^{n}, \overline{\rho}_{2}^{n}, \overline{\rho}_{3}^{n}, \ldots, \overline{\rho}_{N-1}^{n} \right) \]

• Allocate N+1 entries to each vector.
Pseudo-Algorithm

• We now drop the superscript from:

\[
\bar{\rho}_0^{n+1} = 0 \\
\bar{\rho}_i^{n+1} = (1 - \lambda) \bar{\rho}_i^n + (\lambda) \bar{\rho}_{i-1}^n
\]

• And create a two stage version:

Repeat until sufficient time steps taken:

1a) \( \bar{\sigma}_0 = 0 \)

1b) \( \bar{\sigma}_i = (1 - \lambda) \bar{\rho}_i + (\lambda) \bar{\rho}_{i-1} \), \( i=1,...,N-1 \)

2) \( \bar{\rho}_i = \bar{\sigma}_i \), \( i=0,...,N-1 \)

• Note – we just need to evaluate sigmabar using rhobar and then we replace the values in rhobar with the newly computed sigmabar.
Parallel Version

• First divide up the cells into \(\frac{N}{N\text{procs}}\)
• For the purposes of the code you are writing, increase \(N\) until \(N\) is divisible by \(N\text{procs}\).
• i.e. for \(N=12\) (i.e. 12 cells) on 3 processes:

\[
\begin{align*}
\text{Process 0} & \quad \text{Processor 1} & \quad \text{Processor 2} \\
\hline
x_0 & x_1 & x_2 & x_3 & x_4 & x_4 & x_5 & x_6 & x_7 & x_8 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \\
\hline
\overline{\rho}_0 & \overline{\rho}_1 & \overline{\rho}_2 & \overline{\rho}_3 & \overline{\rho}_4 & \overline{\rho}_5 & \overline{\rho}_6 & \overline{\rho}_7 & \overline{\rho}_8 & \overline{\rho}_9 & \overline{\rho}_{10} & \overline{\rho}_{11} \\
\hline
\overline{\sigma}_0 & \overline{\sigma}_1 & \overline{\sigma}_2 & \overline{\sigma}_3 & \overline{\sigma}_4 & \overline{\sigma}_5 & \overline{\sigma}_6 & \overline{\sigma}_7 & \overline{\sigma}_8 & \overline{\sigma}_9 & \overline{\sigma}_{10} & \overline{\sigma}_{11}
\end{align*}
\]
Process Local Storage

• On each process use a logical storage

\[ \bar{\sigma} = (\bar{\sigma}_0, \bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3) \]
\[ \bar{\rho} = (\bar{\rho}_0, \bar{\rho}_1, \bar{\rho}_2, \bar{\rho}_3) \]

• i.e. on procID==0

\[ \bar{\sigma} = (\bar{\sigma}_4, \bar{\sigma}_5, \bar{\sigma}_6, \bar{\sigma}_7) \]
\[ \bar{\rho} = (\bar{\rho}_4, \bar{\rho}_5, \bar{\rho}_6, \bar{\rho}_7) \]

• procID==1

\[ \bar{\sigma} = (\bar{\sigma}_8, \bar{\sigma}_9, \bar{\sigma}_{10}, \bar{\sigma}_{11}) \]
\[ \bar{\rho} = (\bar{\rho}_8, \bar{\rho}_9, \bar{\rho}_{10}, \bar{\rho}_{11}) \]

• procID==2
Updating Sigma

1a) \( \bar{\sigma}_0 = 0 \)

1b) \( \bar{\sigma}_i = (1 - \lambda) \bar{\rho}_i + (\lambda) \bar{\rho}_{i-1} \), \( i=1,...,N-1 \)

2) \( \bar{\rho}_i = \bar{\sigma}_i \), \( i=0,...,N-1 \)

The black and red arrows connect data held locally.

The dashed blue arrow indicates data which needs to be transmitted between processes.

---

The black and red arrows connect data held locally.

The dashed blue arrow indicates data which needs to be transmitted between processes.
(1) Build local version of \( \rho \), \( \rho_0 \), \( \sigma \), \( \sigma_0 \) and \( \lambda \) 
(2) Start looping over time steps 
   a) if(procID != Nprocs-1) 
      \{ \text{MPI_Isend \( \rho \) at end of local vector} \} 
   b) if(procID != 0) 
      \{ \text{MPI_Irecv \( \rho \) for later use.} \} 
      else \{ set first entry of \( \rho \)=0 and first entry of \( \sigma \)=0 } \} 
   c) Compute all but first entry of \( \sigma \) 
   d) If(procID != 0) 
      \{\text{MPI_Wait for Irecv to finish then compute first \( \sigma \) entry} \} 
   e) If(procID != Nprocs-1) 
      \{\text{MPI_Wait for Isend to finish} \} 
   f) Update \( \rho \) from \( \sigma \) (no communication needed) 
   f) Continue to (2)
Details of Communication

Process 0 Isends its copy of rho\textbar 3
Process 1 Irecvs rho\textbar 3 and stores it in a temporary variable.

Process 1 uses the formula to update sigmabar4 using its own rho\textbar 4 and the temporary variable storing Irecv’d rho\textbar 3.
• Serial version of the finite difference code.

• We take a vote to see if people are confident they can finish the parallel version by then 😊
Homework

Q1) Solve the pde analytically on the domain $(-\infty, \infty) \times [0, \infty)$

$$\frac{\partial \rho}{\partial t} + 2 \frac{\partial \rho}{\partial x} = 0$$

$$\rho(x, t=0) = e^{-x^2}$$

Q2) Solve the pde analytically.

$$\frac{\partial \rho}{\partial t} + 2 \frac{\partial \rho}{\partial x} = x - 2t$$

$$\rho(x, t=0) = e^{-x^2}$$

Q3) Solve the pde analytically.

$$\frac{\partial \rho}{\partial t} + 3 \frac{\partial \rho}{\partial x} = -\rho$$

$$\rho(x, t=0) = e^{-x^2}$$

and explain what happens to the density along the characteristics.
Homework cont

Q4) Implement the finite volume approximation of:

\[
\frac{\partial \rho}{\partial t} + 3 \frac{\partial \rho}{\partial x} = 0
\]

\[
\rho(x, t = 0) = e^{-x^2}
\]

By:

Geometry:

\[
x_0 = -4, x_N = 4, x_i = \left(\frac{N - i}{N}\right) x_0 + \left(\frac{i}{N}\right) x_N
\]

Scheme:

\[
\bar{\rho}_i^{n+1} = (1 - \lambda) \bar{\rho}_i^n + (\lambda) \bar{\rho}_{i-1}^n
\]

\[
\lambda = 3 \frac{dt}{dx}
\]

Initial Condition:

\[
\bar{\rho}_i^0 = \rho \left( \frac{x_i + x_{i+1}}{2}, t = 0 \right)
\]

Boundary Condition:

\[
\bar{\rho}_0^0 = 0
\]

Note slight change.
Homework cont

Q4 cont)

a) For N=10,40,160,320,640,1280 run to t=100, with:
   \[ dx = \frac{8}{N} \]
   \[ dt = \frac{dx}{6} \]

b) On the same graph, plot t on the horizontal axis and error on the vertical axis. The graph should consist of a sequence of 6 curves – one for each choice of dx.

c) Comment on the curves.

d) NOTE: For the purposes of this test we define error as:

\[
error^n = \max_{1 \leq i \leq N} \left| \rho_i^n - \rho \left( \frac{x_i + x_{i+1}}{2}, ndt \right) \right|
\]
Homework (parallel part)

Make sure the parallel code gets exactly the same answer as the serial code (to machine precision \( \sim 1e^{-16} \)).

- Use upshot to profile.

For \( N=1024 \) and \( N=10240 \)

- Run on 2, 4, 8, 16 processes and use \( \text{MPI} \cdot \text{Wtime} \) to measure an average time per time step over 400 time steps.

Plot the average time per time step as a function of number of processes for both the 1024 and 10240 cases.

Comment on your graphs (common sense observations).